PROGRAM DÖKÜM EKİ 21

- Changebasic
- Sıralama
- Fonksiyon Tanımlama
- Dosya Sıkıştırıcı
- Isı Kaybı
- Graphic+Text
- 3×12K

COMMODORE
AYLIK BİLGİSAYAR DERGİSİ

Commodore'un Ücretsiz Ekidir

SIRALAMA

Murat SAVAŞ

En yaygın kullanılan işlemler arasında sıralama (verilerin belli şekilde sıralanması) programları vardır. Programımızda olduğu gibi sayıları teker teker ele alıp okumak ve işlemek mutlaka gerekli değildir. Burada dizilere atadığımız değeri büyüklük sırasına göre düzenleriz. Sıralama programları günümüzde öyle önem kazanmıştır ki, bu konuda değişik metodlar bulunmuş ve bu metodlar hakkında pek çok kitap yazılmıştır. Her durum için ideal bir sıralama algoritması yoktur. Ancak bazı metodlar ötekilerden daha iyi iş görür.

Sayı Sıralama:

Programı bloklar halinde incelediğimizde gördüğümüz ilk şey, programın ismi ve değişkenlerin tanımlanmasıdır. Daha sonra ise sayılar adı altında tek boyutlu bir dizinin tanımlanmasını görürüz. Begin anahtar kelimesinden sonra sıralanmasını istediğimiz sayıların (max sabiti kadar) satır atlayarak girilişi, bundan sonraki döngü ile başlatılan blokta ise en büyük sayının bulunuşu vardır. En sonunda sayılar karşılaştırılıp değiş-tokuş yapılır ve döngü yardımıyla ekrana büyükten küçüğe sıralanmış şekilde yazılır. Eğer girilecek değerler sabit olarak tanımlanmış max değerinden büyükse onun değerini büyütebilirsiniz.

Kelime Sıralama:

Bu defa programın amacı bir dizi kelimeyi sıralamaktır. (Programda maksimum 10 kelime uzunluğunda, isterseniz bu değeri değiştirebilirsiniz). Daha önceki sayıların sıralanması programında bahsettiğimiz gibi her zaman için ideal bir sıralama algoritması yoktur. Programın çalışma mantığına gelince: Program kabarcık sıralaması yöntemiyle girilen kelimeleri sıralar. Sıralamaya dizinin en alttaki elemanlarıyla başlanır ve bir üstteki elemanla karşılaştırılır. Eğer alttaki eleman bir üstteki nden küçükse (sudaki hava kabarcığı gibi) yükselir. Bazı program algoritimlerinde ise bu yöntemin tam tersi kullanılır. Buna da "batma sıralaması" denir.

```
PROGRAM KELIMESIRA(INPUT,OUTPUT);
CONST
     A=10;
VAR
     F,B,X:INTEGER;
     C:STRING[10];
     L:ARRAY[1..A] OF STRING[10];
BEGIN
     CLRSCPROGRAM KELIMESIRA(INPUT,OUTPUT);
     WRITECONST
     FOR >     A=10;
          VAR
               F,B,X:INTEGER;
     FOR :     C:STRING[10];
               L:ARRAY[1..A] OF STRING[10];
```

```
        BEGIN
              CLRSCR;
              WRITELN('KELIMELERI GIRINIZ');
              FOR X:=1 TO A DO
                        BEGIN
                        READLN(L[X]);
                        END;
              FOR B:=1 TO A-1 DO
                        BEGIN
    FOR                 FOR F:=B+1 TO A DO
    WRIT                          BEGIN
END.                              IF L[B]>L[F] THEN BEGIN
                                                    C:=L[F];
                                                    L[F]:=L[B];
                                                    L[B]:=C;
                                                    END;
                                  END;
                        END;
              FOR X:= 1 TO A DO
              WRITELN(L[X]);
        END.
PROGRAM SIRALAMA(INPUT,OUTPUT);
CONST
      MAX=5;
VAR
      D,T,K:INTEGER;
      EBN:REAL;
      S:ARRAY[1..MAX] OF REAL;
BEGIN
      CLRSCR;
      WRITELN('ALTALTA SAYILARI GIRINIZ');
      FOR K:=1 TO MAX DO
      READLN(S[K]);
      FOR D:=1 TO MAX -1 DO BEGIN
                              EBN:=S[D];
                              T:=D;
                              FOR K:=D+1 TO MAX DO
                                        IF S[K]>EBN THEN BEGIN
                                                         EBN:=S[K];
                                                         T:=K;
                                                    END;
                              S[T]:=S[D];
                              S[D]:=EBN;
                           END;

      FOR K:=1 TO MAX DO
                        WRITE(S[K]);
END.
```

FONKSİYON TANIMLAMA

Burak KİPER

C-64'de fonksiyon tuşları herhangi bir program içinde bulunan bir seçeneği seçmek için son derece faydalıdır. Birçok bilgisayarda bulunan fonksiyon tuşlarına belirli bir komut grubu yüklemek mümkünken C-64'de bunu yapamıyoruz. İşte programımızın amacı fonksiyon tuşlarını istediğimiz komutları taşıyacak şekilde programlamak.

Programımız çalıştığında istediğiniz bir fonksiyon tuşuna en fazla 30 karakter uzunluğunda bir komut veya komut grubu yükleyebilirsiniz. Bu şekilde kullanılan fonksiyon tuşları hem zamandan kazanmanızı sağlayacak hem de hayatı çekilir hale getirecek.

Mesela F1 tuşuna LIST komutunu yükleyebilirsiniz. Bundan sonra bir programın listesini almak istediğinizde bütün yapmanız gereken F1 tuşuna basmak olacak. Disket kullanıcılarının en büyük dertlerinden biri disketin içeriğini almak için uzun uzun yazmak zorunda oldukları komutlardır. Artık istediğiniz bir fonksiyon tuşuna LOAD"$",S komutunu yükleyerek bu işi tek tuşa indirebilirsiniz.

Programınızı çalıştırmadan evvel muhakkak kasete veya diskete kayıt edin. Program makine dilinde yazıldığı için herhangi bir hata sonunda bütün programı yeniden yazmak zorunda kalabilirsiniz. Programı rahatlıkla yazabilmeniz için bir BASIC yükleyici şeklinde hazırladık. RUN komutu ile çalıştırdıktan sonra NEW yazın. Daha sonra programı aktif hale getirmek için SYS 52224 komutunu verin. Artık fonksiyon tuşlarına istediğiniz komutları vermeye hazırsınız.

F1 tuşundan F6 tuşuna kadar tüm fonksiyon tuşlarını amacınıza uygun şekilde programlayabilirsiniz. Demin de söylediğimiz gibi her fonksiyon tuşuna en fazla 30 karakter uzunluğunda bilgi girebilirsiniz. Programladığınız fonksiyon tuşuna bastığınızda o tuşa verilmiş olan komut sanki klavyeden yazmışsınız gibi ekrana çıkacak ve normal şekilde işleyecektir.

F7 tuşuna bastığınızda fonksiyon tanımlama programımız çalışmaya başlıyor. F8 tuşu ise herhangi bir komut grubunu siler. F7 veya F8 tuşuna bastıktan sonra hangi fonksiyon tuşu üzerinde çalışmak istiyorsanız o tuşa basın. Karşınıza '*' yıldız işareti çıkacak. Şimdi istediğiniz komutları yazarak seçtiğiniz fonksiyon tuşunu programlayabilirsiniz. Yazdığınız komut grubunu bitirmeden evvel RETURN tuşuna basın. Yazmış olduğunuz komutların sonuna sola bakan bir ok eklenecek, bu RETURN tuşunu simgeliyor. Eğer RETURN tuşuna basmazsanız programlamış olduğunuz fonksiyon tuşuna bastığınızda sadece vermiş olduğunuz komut yazılacak, ancak işleme konmayacaktır. Herhangi bir fonksiyon tuşunu programlamayı bitirmek için klavyenin en üst sırasındaki en sol tarafta bulunan sola bakan ok tuşuna basıp tanımlamayı bitirin. Artık fonksiyon tuşunuz tanımlı hale geldi. Fonksiyon tuşuna tanımlamayı yazarken hatalı bir şey yazarsanız bunu silmek için SHIFT ve boşluk tuşlarına beraber basın.

Bu tanımlamada CLR, HOME ve benzeri tuşları kullanırsanız RETURN tuşunda olduğu gibi sola bakan bir ok yazılır. Tanımlamış olduğunuz bir fonksiyonu tekrar aynı yöntemi izleyerek yeniden tanımlayabilirsiniz. Tanımlanmamış veya silinmiş herhangi bir fonksiyon tuşuna bastığınızda bilgisayar bunu göz önüne almayacak.

Makine dili ile ilgilenenler için programımızın kullandığı adresleri verelim:

Program $CC00 (52224) adresinde başlıyor. Yapmış olduğunuz tanımlamalar $CE00 adresinden başlanarak saklanıyor. $0314/5 adresindeki IRQ vektörünü kullanarak programımızı çalıştırıyoruz.

```
170 CO=0:PRINT CHR$(147);"          LUTFEN BEKLEYIN..."
180 REM 8 BOSLUK
190 FOR X=52224 TO 52651 : READ A : POKE X,A : CO=CO+A
200 PRINTCHR$(19);X:NEXT X
210 PRINT
220 IF CO<> 54163 THEN PRINT "DATA SATIRLARI HATALI ! CO=";CO:STOP
230 PRINT"DATA SATIRLARI DOGRU- SAVE EDIP SYS52224 ILE CALISTIRIN."
240 STOP
250 :
260 DATA 76,32,204,80,82,69,83,83,32,75,69,89,32,84
270 DATA 79,32,64,67,76,53,65,82,64,68,69,70,73,78
280 DATA 69,64,1,1,120,169,56,141,20,3,169,204,141,21
290 DATA 3,169,0,168,153,0,206,200,192,0,208,248,88,96
300 DATA 32,159,255,166,198,240,25,202,189,119,2,201,133,144
310 DATA 17,201,141,176,13,198,198,201,136,240,37,201,140,240
320 DATA 6,76,234,204,76,49,234,160,0,32,33,205,160,14
330 DATA 32,93,205,32,108,205,168,169,0,153,0,206,169,13
340 DATA 32,210,255,76,88,204,160,0,32,93,205,160,20,32
350 DATA 93,205,169,13,32,210,255,32,108,205,141,189,204,141
360 DATA 224,204,169,206,141,190,204,141,225,204,169,0,141,30
370 DATA 204,160,1,152,72,32,146,205,170,104,168,138,201,95
380 DATA 240,52,201,160,208,14,192,2,176,3,76,157,204,136
390 DATA 136,169,20,76,205,204,153,0,206,201,32,144,8,201
400 DATA 160,176,6,201,128,144,2,169,95,32,210,255,200,192
410 DATA 32,208,200,169,20,32,210,255,136,76,157,204,136,140
420 DATA 0,206,169,13,32,210,255,76,38,204,56,233,133,168
430 DATA 169,0,192,0,240,7,136,24,105,31,76,240,204,141
440 DATA 7,205,141,78,205,141,27,205,160,0,185,0,206,170
450 DATA 134,193,240,76,162,0,142,30,204,201,11,144,42,72
460 DATA 162,1,189,0,206,32,210,255,232,224,11,208,245,173
470 DATA 30,204,24,105,10,141,30,204,173,27,205,24,105,10
480 DATA 141,27,205,172,30,204,104,56,233,10,76,19,205,170
490 DATA 200,134,198,142,31,204,238,31,204,162,0,185,0,206
500 DATA 157,119,2,232,200,236,31,204,208,243,76,83,204,185
510 DATA 3,204,201,64,240,7,32,210,255,200,76,93,205,96
520 DATA 32,146,205,201,133,144,249,201,140,176,245,201,136,240
530 DATA 241,56,233,133,170,169,42,32,210,255,169,0,224,0
540 DATA 240,7,24,105,31,202,76,134,205,96,173,138,2,72
550 DATA 169,64,141,138,2,32,153,255,32,228,255,201,0,240
560 DATA 241,170,104,141,138,2,138,96
READY.
```

DOSYA SIKIŞTIRICI

Burak KİPER

Sanırım programlarınızı disketten yüklerken, kaybettiğiniz vakitten şikayetçisinizdir. C-64 için yazılmış olan bu programımız ile derdinize biraz olsun çare bulmaya çalışacağız. Programımız BASIC ve diğer tipteki dosyaları özel bir yöntemle kısaltacak, böylece programlarımız diskette daha az yer kaplayacak. Bu bize hem diskette daha fazla yer, hem de programı yüklerken vakit kazandıracak. Programınızı çalıştırdığınızda, programlarınızı kısaltarak kaydedebilir veya kısaltılmış programlarınızı yükleyebilirsiniz.

Programımız $C200 ve $C617 adresleri arasında yer alıyor. Programı diskete sakladıktan sonra, LOAD"DOSYA SIKIŞTIRICI",8,1 ile yükleyebilirsiniz. Programın yüklenmesi bittikten sonra NEW komutu vererek önemli BASIC belirteçlerinin sıfırlanmasını sağlayın.

Basic Programları Kısaltma

Programımız iki değişik şekilde kullanılabilir. İlk yöntem ile BASIC programlarınızı sıkıştırılmış şekilde yükleyebilir veya saklayabilirsiniz. Programımızı çalıştırmak için öncelikle yükleyin ve SYS50600 yazdıktan sonra RETURN tuşuna basın. BASIC bir programı sıkıştırılmış şekilde saklamak için o programı hafızaya yükledikten sonra SAVE yazıp RETURN tuşuna basın. Bilgisayar sizden dosya adı girmenizi istediğinde bir isim vererek RETURN tuşuna basın. Programınız diskete verdiğiniz isimde sıkıştırılmış olarak kayıt edilecek. Sıkıştırılmış bir BASIC programı yüklemek için LOAD yazıp RETURN tuşuna basın ve yüklemek istediğiniz dosya ismini girin.

BASIC programlarınızı normal şekilde yüklemek veya saklamak isterseniz, LOAD"program-ismi",8,1 şeklinde alıştığımız yöntemi kullanabilirsiniz.

Veri Sıkıştırma

Dosya sıkıştırma programımız, herhangi bir hafıza bölgesinin de sıkıştırılarak saklanmasını sağlamakta, bu hafıza bölgeleri, hi-res grafik ekranı, sprite şekilleri veya her tür çeşitten veri olabilir. Sıkıştırarak sakladığınız verileri de tekrar normal hale getirebilirsiniz. DOSYA-1 adlı küçük program size hafızanın belirli bir bölgesinin nasıl saklanacağını göstermekte.

Programımız SYS komutu ile aktif hale gelmeden evvel, saklamak istediğimiz hafıza bölgesinin başlangıç ve bitiş adreslerini 679-690 ve 681-682 numaralı adreslere POKE ediyoruz. Bundan başka dosya ismi için ASCII karakterleri 832 numaralı adresten başlayarak yerleştirmek zorundayız. Programda 10006 numaralı satırda dosya isminin sonuna P, W ve buna da CHR$(0) eklendiğine dikkat edeceksiniz. DOSYA-2 adlı programda sıkıştırılarak saklanmış bir dosyayı yükleyecektir.

Bu yordam da, saklama yordamına benzemekte, fakat bu sefer sadece programın başlama adresini vermemiz yeterli olmakta.

Programın Çalışma Sistemi

Elbette bu programın nasıl çalıştığını bilmeden kullanabilirsiniz, fakat biz programın nasıl çalıştığının kısa bir açıklamasını yapalım. Programın çalışma şeklindeki temel amaç devamlı olarak tekrarlanan bilgilerin ekonomik bir şekilde saklanmasıdır. Hemen hemen birçok veride devamlı olarak tekrarlanan yerlere rastlarız. Mesela birçok metinlerde sesli harfler oldukça sık kullanılırlar. Bizim programımız bu sık sık kullanılan bilgileri alışılmıştan daha az bir yerde saklayarak yer kazanmayı amaçlamakta. Bunu anlamak için bilgisayarın verileri nasıl

sakladığına bir bakalım.

C-64 bütün verileri byte'lardan oluşmuş seriler halinde saklar. Bir byte sekiz bit'ten oluşmuştur, her bit ise 1 veya 0 değerini almaktadır. İkilik düzende 00000000 sayısı 0'a 11111111 ikilik sayısı da onluk düzende 255 sayısına eşittir. Bu durumda bir byte 0 ile 255 sayıları arasında bir değer alabilir. Her byte sekiz bit'ten oluştuğu için, bilgisayar bir byte'ın nerede bitip diğerinin nerede başladığını anlamak için bitleri sayar. Sekizinci bite ulaştığında bir sonra gelecek bitin bir sonraki byte'ın ilk biti olacağını anlar.

Bu arada şunu unutmayalım, bir byte değeri kullanıldığı yerlere göre değişik anlamlara gelebilir, mesela bir kelime işlem programının metin kısmında geçen ASCII 65 değeri 'a' harfine karşılık gelirken, bu değer bir sprite tanımlamasında bir dizi noktaya karşılık gelecek, makine dili programında geçiyorsa EOR komutunu ifade edecek.

Bizim programımız çok sık kullanılan kodları sekiz bit yerine daha az sayıda bit'e sığdırmaya çalışmakta, bunu şu şekilde açıklayalım: Mesela bir kelime işlem programı ile yazılmış bir metni sıkıştırarak saklamak istiyorsunuz, programınız saklama işlemine başlamadan evvel saklanacak veriyi tarayarak en sık geçen yedi karakteri belirleyecek. Belirlediği değerlere göre bir kodlama tablosu oluşturulduktan sonra bu tablo hafızaya olduğu gibi disketteki dosyanın başına da yerleştirilir. En çok kullanılan değerin saklanması için sadece bir bit yeterli olmakta, SAVE işlemi sırasında bu kodlar normal sekiz bitlik gösterimin yerini almakta. Programı yükleyeceğimiz zaman önce kod tablosu okunup program bu kodlama tablosuna göre normale çevrilerek yüklenir.

```
10000 REM SIKISTIRILMIS SAVE SA=SAKLAMANIN BASLANGICI:EA=SAKLAMANIN SONU
10001 REM FL$=DOSYA ADI
10002 POKE 680,SA/256:POKE679,SA-PEEK(680)*256
10004 POKE 682,EA/256:POKE 681,EA-PEEK(682)*256
10006 FL$=FL$+",P,W"+CHR$(0):FOR I=1 TO LEN(FL$)
10008 POKE 831+I,ASC(MID$(FL$,I,1)+CHR$(0)):NEXT
10010 SYS 50689:RETURN
READY.
```

ISI KAYBI HESABI

Doç.Dr. Fethi ERALP

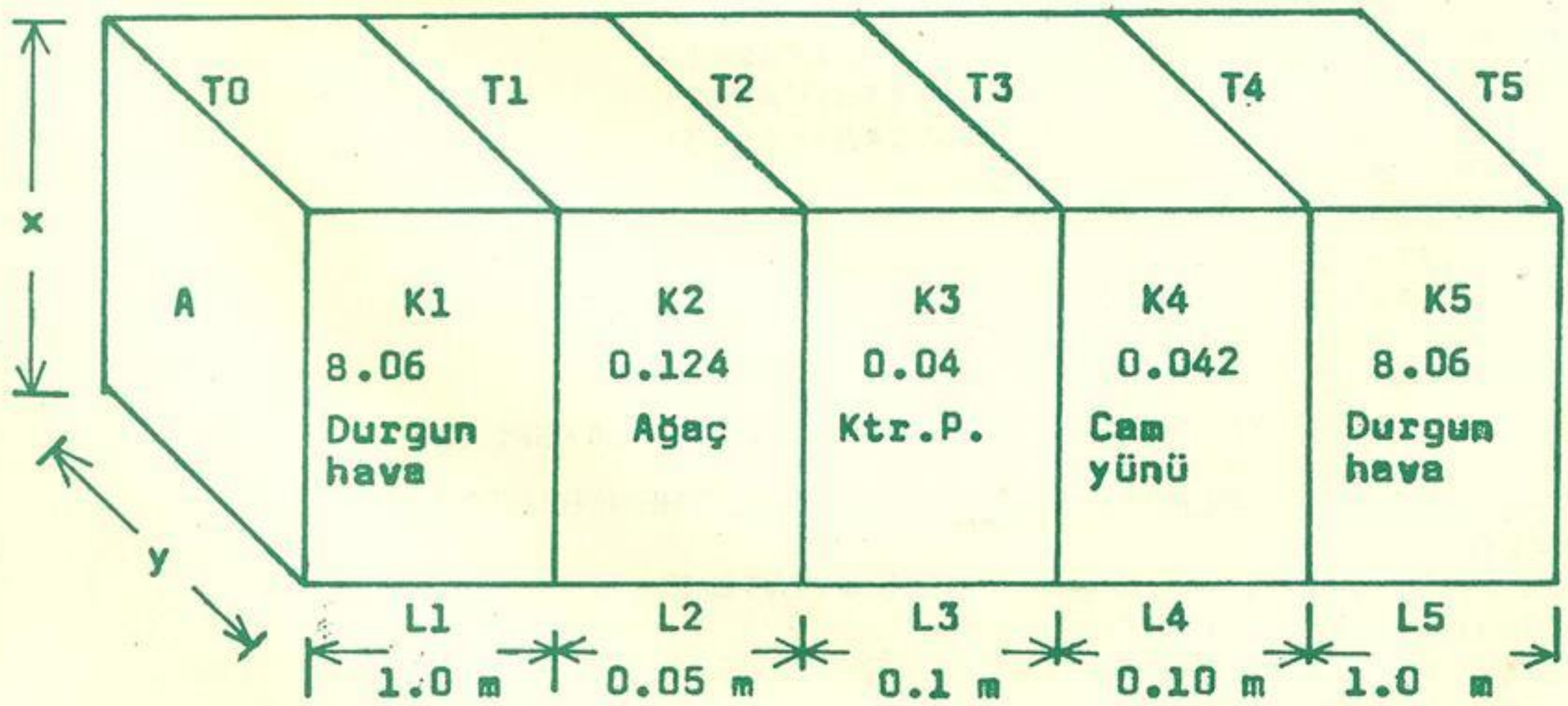
Isı kaybı hesabı programı Commodore-64 için hazırlanmıştır. Açıklama: Bu program bir gemide bulunan soğutma sistemini meydana getiren değişik odaların tavanlarından, tabanlarından, bölmelerinden ve antreden oluşan ısı kayıplarının bulunmasını amaçlıyor. Tavanlar için bulunan değer 1.3 ile ve bölümler için bulunan değer 1.1 ile çarpılır. Aşağıda ele alınan örnek yalnız tavan içindir. Et odasının sıcaklığı -8 C ve bu odanın dış kısmındaki sıcaklık ise 45 C alınmıştır.

Şekil 1'de tavan levhasının hangi malzemelerden oluştuğu, tavan alanı, malzemelerin kalınlıklarını ve k değerleri ile iç ve dış sıcaklıkları görülmektedir.

Herhangi bir levhanın hangi malzemelerden yapıldığı, bu malzemelerin kalınlıkları ve k değerleri bilinirse bu levhadaki ısı kaybı Şekil 2'de verilen programla kolayca bulunur.

Şekil 3'de, Şekil 1'deki değerlere göre bilgisayardan alınan sonuçların print'i görülmektedir.

Bir soğutma tesisinin tam olarak (ayrıntılı biçimde) hesabının nasıl yapılacağı ve bu ısıyı alacak kompresörün kapasitesinin ne olacağı; İTÜ Gemi İnşaatı ve Deniz Bilimleri Fakültesi yayını olan ve Fethi Eralp tarafından yazılmış "Gemi Yardımcı Makineleri-2" kitabında bulunabilir.



Şekil - 1

```
READY.
 1 REM          ******************
 2 REM          *                  *
 3 REM          *    ISI KAYBI     *
 4 REM          *     HESABI       *
 5 REM          *                  *
 6 REM          ******************
 7 REM
 10 DIM Z(20):POKE53280,PEEK(53281)
 12 FA$(1)="(TAVAN)       1.3":FA$(2)="(YAN PERDE) 1.
 1":FA$(3)="(BOLME)       1.0"
 13 FA$(0)="ISI KAYBI FAKTORU"
 15 PRINTCHR$(147):INPUT"YAZICI CIKISI (E/H)";YA$:I
FYA$="E"THENOPEN4,4
 20 PRINTCHR$(147)SPC(8)"*** ISI KAYBI HESABI ***"
 25 PRINTCHR$(17):INPUT"X=";X:IFYA$="E"THENPRINT#4,
"   X =";XCHR$(13)
 30 PRINTCHR$(17):INPUT"Y=";Y:IFYA$="E"THENPRINT#4,
"   Y ="YCHR$(13)
 35 A=X*Y
 40 FORI=1TO69:J=I*2-2
 50 PRINTCHR$(17)"L"I"=";:INPUTZ(J):IFZ(J)=0THEN80
 55 IFYA$="E"THENPRINT#4,"L"I"="Z(J)
 60 PRINT"K"I"=";:INPUTZ(J+1):PRINT
 65 IFYA$="E"THENPRINT#4,"K"I"="Z(J+1)CHR$(13)
 70 NEXTI:I=I-1:J=J+2
 80 IFI=1THEN40
 100 PRINTCHR$(17);:INPUT"T 0 =";T:IFYA$="E"THENPRI
NT#4,"T 0 ="T
 105 PRINT"T"I-1;:INPUTU:IFYA$="E"THENPRINT#4,"T"I-
1"="UCHR$(13)CHR$(13)
 110 Q=0:FORI=0TOJ-1STEP2
 120 Q=Q+Z(I)/Z(I+1)/A
 130 NEXTI:Q=(T-U)/Q
 140 PRINTCHR$(147)CHR$(32)FA$(0)CHR$(13):IFYA$="E"
 k4,FA$(0)" =";
 150 PRINTCHR$(18)" 1 "CHR$(146)FA$(1)
 160 PRINTCHR$(18)" 2 "CHR$(146)FA$(2)
 170 PRINTCHR$(18)" 3 "CHR$(146)FA$(3)
 180 GETC$:IFC$=""THEN180
 190 IFC$="1"THENM=1.3:GOTO220
 200 IFC$="2"THENM=1.1:GOTO220
 210 IFC$<>"3"THEN140
 215 M=1
 220 C=VAL(C$)
 222 IFYA$="E"THENPRINT#4,RIGHT$(FA$(C),4)SPC(3)LEF
T$(FA$(C),12)CHR$(13)
 224 PRINT"CEVAPLAR":PRINT:IFYA$="E"THENPRINT#4,"CE
VAPLAR"CHR$(13)
 225 PRINTCHR$(147)CHR$(29)CHR$(29)CHR$(29)"Q="INT(
Q*M*1E3)/1E3"[KCAL/H)]"
 227 IFYA$="E"THENPRINT#4,"   Q ="INT(Q*M*1E3)/1E3;C
HR$(16)"15";"[KCAL/H]"
 230 FORI=0TOJ-2STEP2
```

```
240 T=T-Z(I)*Q/Z(I+1)/A:PRINT"T"I/2+1"="INT(T*1E3)
/1E3" SANTIGRAT DERECE"
245 IFYA$<>"E"THEN260
247 PRINT#4,"T"I/2+1"="INT(T*1E3)/1E3;CHR$(16)"15"
;"SANTIGRAT DERECE"
260 NEXTI:IFYA$="E"THENPRINT#4,CHR$(13)CHR$(13)CHR
$(13)
265 GETC$:IFC$=""THEN265
270 GOTO25
```

Şekil - 2

```
   X = 14.25

   Y = 1

L 1 = 1
K 1 = 8.06

L 2 = .05
K 2 = .124

L 3 = .1
K 3 = .04

L 4 = .1
K 4 = .042

L 5 = 1
K 5 = 8.06

T 0 = 45
T 5 =-8
```

```
CEVAPLAR

   Q = 177.47    [KCAL/H]
T 1 = 43.811     SANTIGRAT DERECE
T 2 = 39.948     SANTIGRAT DERECE
T 3 = 15.998     SANTIGRAT DERECE
T 4 =-6.812      SANTIGRAT DERECE
T 5 =-8          SANTIGRAT DERECE
```

Şekil - 3

```
ISI KAYBI FAKTORU = 1.3    (TAVAN)
```

GRAPHİC+TEXT

Okan YILMAZ

Bazı macera oyunlarında (advanture games) veya grafik kullanım programlarında (graphic utulities) ekranın bir kısmının high-resolution grafik ekranı, diğer kısmının normal yazı ekranı olarak kullanıldığını görmüşsünüzdür.

Bunun nasıl gerçekleştiğini hiç merak ettiniz mi? Daha da ötesi, bu özelliği kendi programlarınızda kullanmak ister miydiniz? Yukarıdaki soruların en az birine evet cevabı vermiş iseniz bu program tam size göre...

Programı yazmak için yine MDE'-ye ihtiyacınız olacak. Sorulacak sorulara (program adı: GRF+TEXT, başlama adresi:$0801, bitiş adresi:$0901) cevaplarını verin. Programı yazıp, kaydettikten sonra tekrar yükleyin ve RUN komutunu verin. Ekran çerçeve renginin (border) bir an için yanıp sönerek size işaret verdiğini göreceksiniz. Artık programımız kullanıma hazırdır.

SYS 40784 komutu verdiğinizde ekranın üst yarısının hi-res grafik ekranı, alt yarısının ise normal yazı ekranı olduğunu göreceksiniz. Eğer bu komutu ekranın üst yarısında yazmış iseniz şu anda cursör'ü göremiyor olabilirsiniz. Ama meraklanmayın CRSR aşağı tuşuna bir süre basarak tekrar görebilirsiniz onu.

Grafik ekranını silmek için SYS 40914 komutunu, renklendirmek için ise POKE 780, sayı:SYS 40936 komutunu kullanabilirsiniz. Burada sayı: zeminin rengi+16×çizgi rengi'-dir.

Ekrandaki grafik sayısını kapatmak ve tamamen yazı ekranına dönmek için ise SYS 40861 komutunu kullanabilirsiniz. Eğer ekranınızın tamamen grafik ekranı olmasını istiyorsanız SYS 40882 komutu, bu durumdan tekrar yazı ekranına dönmek için ise SYS 40898 Komutu emrinizdedir. Ekrandaki grafik sayfasının sınırlarını değiştirmek isterseniz POKE 1021,grf. başl. satırı ve POKE 1022, grf. bitiş satırı komutları ile başarabilirsiniz. Bu registerlerin ilk değerleri sırasıyla 1,161'dir. Birkaç deneme ile amacınıza uygun satır numaralarını bulabilirsiniz.

Son olarak grafik hafızası olarak Bank2'nin ikinci yarısını ($A000-BF3F), video ram olarak da $8400-$8800 arasını kullandığımızı söyleyerek programla birlikte kullanacağınız utulity programlarında veya kendi rutinlerinizde bu hususlara dikkat etmeniz gerektiğini vurguluyorum. Ayrıca programımızın PLASTİK GRAFİK (COMMODORE Dergisi, 4. sayı) programı ile tam uyumlu çalıştığını söylemek istiyorum. Birlikte çalıştırmak için önce PLASTİK GRAFİK programını çalıştırın. Daha sonra GRF+TXT programını yükleyin. Her iki programın komutları da aynen hizmetinizde.

Geldik programımızın açıklanmasına;

Programımız tarama interrupti tekniği (Raster interrupt technique) ile gerçekleştirilmiştir. Bildiğimiz gibi C-64 saniyede 60 kere işlettiği programı bırakarak başka bir programı işletir. Bu işletilen programın bir kısmı ekranın taranarak görüntünün yinelenmesi görevini yerine getirir. Eğer tarama satırı (raster line) belli bir satıra geldiğinde grafik ekranını açıp, diğer bir satıra ulaşınca da kapatırsak bu işlem saniyede 60 kez tekrarlandığı için gözümüz bundan rahatsız olmaz. Böylece amacımıza ulaşmış oluruz.

Tarama satırının bulunduğu adres $DO12'dir. Grafik başlama ve bitiş satırlarını bu adrese yerleştiriyoruz. C-64'e tarama satırı interruptı gerçekleştirmek istediğimizi bildirmek için VIC'in 26 numaralı yazmacına (interrupt mask register, IMR) # % 10000001 değerini atamamız gerekiyor. Ayrıca IRQ vektörleri $314/$315'i bizim programımızı işaretleyecek şekilde değiştiriyoruz. Saniyede 60 kez bizim rutinimize sapınca interrupt oluşma nedenleri yazmacı kontrol edilir. Eğer bu register (IRR) interrupta izin verirse tarama satırın değerine göre grafik ekranı açılır veya kapatılır.

Grafik ekranının açılması, kapatılması, temizlenmesi, renklendirilmesi rutinlerini açıklamaya gerek görmüyorum. Bu konuda bilgiye ihtiyaç duyan arkadaşlar için en yakın ve açık kaynağın sayın Tuna Ertemalp'in Commodore 64 ile Grafik yazı dizisi olduğunu belirtmeme gerek yok sanırım.

Şimdiden iyi kullanımlar dilerim...□

```
100 ;=============================
101 ;-      GRAPHIC + TEXT       -
102 ;-                           -
103 ;- DESIGNED BY  OKAN YILMAZ -
104 ;=============================
105 ;
106 .EQ VIC=53248     ;(=$D000)
107 .EQ CIA=56576     ;(=$DD00)
108 .EQ GRAPHIC=$A000;GRAFIK SAYFASI
109 .EQ GRFEND=$BFFF  ;GRAFIK BITISI
110 .EQ VRAM=$8400    ;VIDEO RAM
111 .EQ ADR=$FB
112 .EQ ACU=$30C      ;RENK BILGISI BU ADRESTE
113 .EQ IRQVEC=$314   ;IRQ VEKTORU
114 .EQ IRQOLD=$EA31 ;IRQ ROM RUTINI
115 .EQ GETREG=$EA81 ;REGISTERLERI YIGINDAN ALAN RUTIN
116 .EQ RASTER=VIC+18;TARAMA SATIRI
117 .EQ IRR=VIC+25    ;INTERRUPT KAYNAGI
118 .EQ IMR=VIC+26    ;INTERRUPT MASKESI
119 .EQ LINE1=$3FD    ;GRF. BASLANGICI
120 .EQ LINE2=$3FE    ;GRF. BITISI
121 ;
122 .BA $3F50         ;BASLAMA ADRESI
123 ;
124 ;  TARAMA INTERRUPTINA HAZIRLIK
125 ;
126 INIT SEI;INTERPUPT'I ONLE
127  LDALINE1;GRF. BASLANGICINI AL
128  STARASTER;TARAMA SATIRINA YERLESTIR
129  LDARASTER-1;TARAMA SATIRI 9. BITINI
130  AND#127;%01111111 ILE SONDUR VE
131  STARASTER-1;YERLESTIR
132  LDA#129;%10000001 DEGERI ILE
133  STAIMR;TARAMA INTERRUPT'INI MASKELE
134  LDA#<(TEST);INTERRUPT VECTORUNU
135  LDY#>(TEST);DUZENLE
136  STAIRQVEC;VE SAKLA
137  STYIRQVEC+1
138  CLI;INTERRUPTI SERBEST BIRAK
139  RTS
140 ;
141 TEST LDAIRR;INTERRUPT KAYNAGINI OKU
142  STAIRR;VE SAKLA
143  AND#1;(%00000001) TARAMA INTERRUPTI MI?
144  BNERASTINT
145  JMPIRQOLD;HAYIR,NORMAL INTERRUPT RUTININE
146 ;
147 ;     TARAMA INTERRUPTI
148 ;
149 RASTINT LDARASTER;EVET.TARAMA SATIRI
150  CMPLINE2;GRF. KAPATMA SATIRI MI?
151  BCSSECOND;EVET,SECOND'A ATLA
152  LDALINE2;HAYIR,GRF. KAPATMA SATIRINI
153  STARASTER;TARAMA SATIRINA KOY
```

```
154  JSRGRFON;GRF. AC
155  JMPGETREG;REGISTERLERI GERI ALMA RUTININE ATLA
156 SECOND LDALINE1;GRF.ACMA SATIRINI
157  STARASTER;TARAMA SATIOINA KOY
158  JSRGRFOFF;GRF. KAPAT
159  JMPGETREG;REGISTERLERI ALMA RUTININE ATLA
160 ;
161 ;TARAMA INTERRUPTININ KAPATILMASI
162 ;
163 RASTOFF SEI;INTERRUPTI ONLE
164  LDA#0;INTERRUPT MASKESINI
165  STAIMR;DUZENLE
166  JSRGRFOFF;GRF. KAPAT
167  LDA#<(IRQOLD);IRQ VEKTORLERINI
168  LDY#>(IRQOLD);ORIGIMAL IRQ
169  STAIRQVEC;ADRESINE
170  STYIRQVEC+1;GUNLE
171  CLI;INTERRUPTA IZIN VER
172  RTS
173 ;
174 ; HI-RES GRAFIK ACMA RUTINI
175 ;
176 GRFON LDA#59
177  STAVIC+17
178  LDA#143
179  STACIA
180  LDA#$18
181  STAVIC+24
182  RTS
183 ;
184 ; HI-RES GRAFIK KAPAMA RUTINI
185 ;
186 GRFOFF LDA#27
187  STAVIC+17
188  LDA#151
189  STACIA
190  LDA#21
191  STAVIC+24
192  RTS
193 ;
194 ; HI-RES GRAFIK SILME RUTINI
195 ;
196 CLEAR LDA#>(GRAPHIC)
197  LDY#<(GRAPHIC)
198  STAADR+1
199  STYADR
200  LDX#32
201  TYA
202 LOOP1 STA(ADR),Y
203  INY
204  BNELOOP1
205  INCADR+1
206  DEX
207  BNELOOP1
```

```
208  RTS
209 ;
210 ;  HI-RES GRAFIK RENK RUTINI
211 ;
212 COLOR LDAACU;RENK BILGISI $30C (780) ADRESINDEN ALINIYOR
213  LDX#>(VRAM)
214  LDY#<(VRAM)
215  STXADR+1
216  STYADR
217  LDX#4
218 LOOP2 STA(ADR),Y
219  INY
220  BNELOOP2
221  INCADR+1
222  DEX
223  BNELOOP2
224  RTS
READY.
```

3 x 12K

Okan YILMAZ

Commodore 64'ünüzün içinde üç adet, birbirinden bağımsız bilgisayar olsun ister miydiniz? İşte bu kısacık program, BASIC RAM'ını 12K'lık bloklar halinde üç ayrı bölgeye ayırıyor. Böylece birbirinden bağımsız üç adet 12K'lık COMMODORE elde etmiş oluyorsunuz. Durun canım... Hemen 12K bellek çok yetersiz diye sızlanmayın. COMMODORE dergisinde yayınlanan en uzun BASIC programlar bile 10K civarında. Birçok kullanımda usta programcıya 12K yeter de artar bile...

Bu program sayesinde bellekte aynı anda birbirinden bağımsız üç program bulundurabilir, bu programlar üzerinde ayrı ayrı çalışabilir, birbirlerini hiç etkilemeden yükleme ve kaydetme işlemlerini gerçekleştirebilirsiniz. Örneğin, yaptığınız bir programı COMMODORE 1'de saklarken, COMMODORE 2'de yeni yaptığınız altprogramları test edebilir ve aynı anda COMMODORE 3'te de disket directory'si veya başka bir programı bulundurabilirsiniz.

Programı yazabilmek için MDE'yi yükleyin ve çalıştırın. Sorulacak sorulara (Program adı: 3 x 12K, Başlama adresi: $0801, Bitiş adresi: $090E) cevaplarını verin. Programı yazıp kasete veya diskete kaydettikten sonra tekrar yükleyin ve 'RUN' ile çalıştırın. Ekranın temizlendiğini ve 'READY' mesajını göreceksiniz. Artık 12K'lık COMMODORE'larınız hazır durumdadır. COMMODORE'lar arasında geçiş yapabilmek için kullanacağınız komut: !COM veya !C'dir. Denediğinizde '1, 2 veya 3 giriniz.' mesajı ile karşılaşabilirsiniz. Örneğin 3'e bastığınızda 'COMMODORE 3 AKTİF DURUMDA.' cevabını alırsınız. Herhangi bir işlem için başka bir bloğa geçmeniz gerektiğinde yine, !C komutuna başvurmanız gerekiyor. O halde hangi blokta bulunduğunuzu öğrenmek için !TEST veya sadece !T komutunu kullanın.

Şimdi de programımızın içeriğine bir göz atalım:

İlk bölümde BASIC komutu yürürlüğe koyma adresini belirleyen vektörü değiştiriyoruz. Böylece kendi dizayn edeceğimiz yürürlüğe koyma rutininde özel komutlarımızı tanımlama imkanı bulacağız. Bu bölümde ayrıca her bir bloğun ilk üç adresini sıfırlıyor ve ROM'daki NEW rutinine atlıyoruz.

154 ile 158 satırları arasında karşılaşılan karakterin "!" olup olmadığı kontrol ediliyor. Eğer "!" ile karşılaşmış isek bu bizim özel komutumuzdur, aksi halde ROM'daki yürürlüğe koyma rutinine atlıyoruz.

159 ile 163 Satırları arasında "!" karakterinden sonra gelen harf değerlendiriliyor. Eğer bu bir "C" ise !C komutu, "T" ise !T komutu yürürlüğe konuluyor.

164-175 satırları arasında !T komutu görüldüğü zaman yürürlüğe konulacak rutin bulunuyor. Burada hangi blok aktif ise test ediliyor ve ekrana bir mesaj yollanıyor.

177. satırdan program sonuna kadar ise !C komutu ile karşılaşıldığında yürürlüğe konulacak ana rutin bulunuyor. Bildiğiniz gibi COMMODORE 64'ün BASIC RAM'ını organize eden vektörler şunlardır:

$2B-$2C 43-44
BASIC program başlangıç vektörü

$2D-$2E 45-46
BASIC prg.bitişi/Değişken başl.vek.

$2F-$30 47-48
Değişken bitiş vek./Boyut başl.vek.

$31-$32 49-50
Boyut bitiş vektörü

$37-$38 55-56
BASIC RAM sonu vektörü

Bu rutin çağırıldığı zaman önce hangi blokta ise o bloğa ait yukarıdaki vektörleri tabloya kaydediyor. Böylece bir daha çağırıldığı zaman bu vektörlere eski değerlerini yükleyebilecek ve sanki hiçbir şey olmamışcasına görevine kaldığı yerden devam edebilecektir. Sonra bizim tercihimizi (1, 2 veya 3) alarak tablodan vektörlere ait yeni değerlere ulaşıyor. Bu değerleri vektörlere yerleştirdiğimizde herşeyin hazır olduğuna dair bir de mesaj yollayarak rutin sona eriyor. İşte hepsi bu kadar...

Programın Assembler Listing'ini inceledikten sonra tablolarda yapacağınız uygun değişikliklerle BASIC RAM'ı 2 x 18K veya 4 x 9K yapabilirsiniz değil mi?

İyi kullanımlar... □

```
100 ;============================
101 ;-          3*12K            -
102 ;-                           -
103 ;-   OKAN YILMAZ / ANKARA    -
104 ;============================
105 ;
106 .EQ BSCSTRT = $2B     ;BASIC BASLANGIC ADRESI VECTORU
107 ;
108 .EQ  BSCEND = $37     ;BASIC BITIS ADRESI VECTORU
109 ;
110 .EQ VARSTRT = $2D     ;DEGISKEN BASLANGIC ADRESI VECTORU
111 ;
112 .EQ ARRYSTR = $2F     ;BOYUT BASLANGIC ADRESI VECTORU
113 ;
114 .EQ ARRYEND = $31     ;BOYUT BITIS ADRESI VECTORU
115 ;
116 .EQ     GET = $FFE4   ;KERNAL GET RUTINI
117 ;
118 .EQ     MAX = 3       ;MAX. SECENEK
119 ;
120 .EQ  CHRGET = $73     ;ZERO PAGE KARAKTER ALMA RUTINI
121 ;
122 .EQ  CHRGOT = $79     ;CHRGET RUTININ DEVAMI
123 ;
124 .EQ    EXEC = $308    ;BASIC KOMUT YURURLUGE KOYMA VECTORU
125 ;
126 .EQ EXECOLD = $A7E7   ;BASIC KOMUT YURURLUGE KOYMA ROM RUTINI
127 ;
128 .EQ   READY = $A474   ;READY MESAJI YOLLAYAN ROM RUTINI
129 ;
130 .EQ STRGOUT = $AB1E   ;STRING YAZAN ROM RUTINI
131 .EQ   PRINT = $FFD2   ;KARAKTER YAZAN ROM RUTINI
132 ;
133 .EQ     NEW = $A644   ;ROM NEW RUTINI
134 ;
135 .EQ    COM1 = 14335   ;COMPUTER 1 BITIS ADRESI
136 ;
137 .EQ    COM2 = 26623   ;COMPUTER 2 BITIS ADRESI
138 ;
139 .BA $C000 ;BASLANGIC ADRESI
140 ;
141 INIT LDA#<(TEST);BASIC KOMUTU
142  LDY#>(TEST);YURURLUGE
143  STAEXEC;KOYMA ADRESLERI
144  STYEXEC+1;DUZENLENIR
145  LDX#3
146  LDA#0;COM. 1 VE COM.2 NIN
147 ZERO STACOM1,X;BITIS ADRESLERI
148  STACOM2,X;SIFIRLANIR
149  DEX
150  BNEZERO
151  JSRNEW
152  RTS
153 ;
```

```
154 TEST JSRCHRGET;KAREKTER AL
155  CMP#"!";OZEL ISARETIMIZ MI?
156  BEQFOUND;EVET
157 NOCMD JSRCHRGOT;HAYIR,O HALDE
158  JMPEXECOLD;ROM RUTININE ATLA
159 FOUND JSRCHRGET;KARAKTER AL
160  CMP#"C";"C" MI?
161  BEQFOUNDC;EVET
162  CMP#"T";"T" MI?
163  BNENOCMD:HAYIR KOMUT BULUNAMADI
164 FOUNDT LDA#<(CMPTR);"T" KOMUTU BULUNDU
165  LDY#>(CMPTR);MESAJ ADRESINI A/Y YUKLE
166  JSRSTRGOUT;MESAJI YAZ
167  LDACHOOSE;AKTIF COM. NO'SUNU AL
168  CLC;ASCII KODUNA CEVIR
169  ADC#48
170  JSRPRINT;EKRANA YAZ
171  LDA#<(ACTIV);SONRAKI MESAJ ADRESINI
172  LDY#>(ACTIV);A/Y MODUNDA AL
173  JSRSTRGOUT;MESAJI YAZ
174  JMPREADY;READY MESAJI YOLLA
175 ;
176 FOUNDC LDXCHOOSE:ESKI VECTORLERI
177  LDAVARSTRT;TABLODA
178  STATABLE,X;SAKLA
179  LDAVARSTRT+1
180  STATABLE2,X
181  JSRMESAJ;SECIM MESAJINI YOLLA
182 GETLOOP JSRGET;KARAKTER ALINANA
183  AND#15;KADAR BEKLE
184  BEQGETLOOP
185  CMP#MAX+1;KARAKTER 1,2 VEYA 3 DEGILSE
186  BCSGETLOOP;TEKRAR GET RUTININE
187  TAX
188  STXCHOOSE;YENI PARAMETREYI SAKLA
189  LDACHOOSE,X;VECTORLERI
190  STABSCSTRT+1;YENI
191  LDACHOOSE+1,X;PARAMETREYE
192  STABSCEND+1;GORE
193  LDATABLE,X;DUZENLE
194  STAVARSTRT
195  STAARRYSTR
196  STAARRYEND
197  LDATABLE2,X
198  STAVARSTRT+1
199  STAARRYSTR+1
200  STAARRYEND+1
201  JMPFOUNDT
202 CHOOSE .BY1,8,$38,$68
203 TABLE .BY$98,3,3
204 TABLE2 .BY3,8,$38,$68
205 ;
206 MESAJ LDA#<(MES);MESAJ RUTINI
207  LDY#>(MES)
```

```
208  JSRSTRGOUT
209  RTS
210 MES .BY$0D
211 .TX " 1,2 VEYA 3 GIRINIZ."
212 .BY $0D,0
213 CMPTR .TX"COMMODORE "
214 .BY 0
215 ACTIV .TX" AKTIF DURUMDA."
216 .BY 0
READY.
```

1541

```
 PROGRAM   : DISKMON          0801 14A1
-------------------------------------

0801 : 0B 08 C1 07 9E 32 30 36   0A
0809 : 31 00 00 00 A9 2C A0 08   C9
0811 : 85 5F 84 60 A9 A1 A0 14   C5
0819 : 85 5A 84 5B A9 75 A0 CC   BA
0821 : 85 58 84 59 20 BF A3 4C   46
0829 : 00 C0 00 A9 80 8D 8A 02   61
0831 : 78 A9 6A 8D 14 03 A9 C0   4C
0839 : 8D 15 03 A9 00 8D 12 D0   9D
0841 : AD 11 D0 29 7F 8D 11 D0   1A
0849 : A9 81 8D 1A D0 58 A9 00   D0
0851 : 8D 71 C9 A9 01 8D 21 D0   E1
0859 : A2 D8 A0 C9 20 3A C9 A2   09
0861 : 0D A9 2E 20 E4 C7 A9 00   06
0869 : 8D 6D C9 20 CF FF C9 2E   A4
0871 : F0 F9 C9 20 F0 F5 A2 13   44
0881 : A9 C0 48 A9 33 48 BD 87   4D
0889 : C9 48 BD 9B C9 48 60 CA   4F
0891 : 10 E6 4C 34 C0 AD 19 D0   2D
0899 : 8D 19 D0 10 30 AD 12 D0   43
08A1 : C9 64 B0 0C A9 08 8D 20   9B
08A9 : D0 A9 64 8D 12 D0 D0 1A   38
08B1 : C9 D2 B0 0C A9 03 8D 20   EA
08C9 : 20 D0 4C BC FE A9 C0 48   CD
08D1 : A9 E6 48 08 48 48 48 AD   44
08D9 : 0D DC 58 4C 31 EA AD 71   F8
08E1 : C9 D0 11 78 A9 6A 8D 14   B2
08E9 : 03 A9 C0 8D 15 03 A9 81   B6
08F1 : 8D 1A D0 58 60 AD 71 C9   97
08F9 : D0 FA 78 A9 31 8D 14 03   6F
0901 : A9 EA 8D 15 03 A9 01 8D   C2
0909 : 20 D0 A9 00 8D 1A D0 58   99
0911 : 60 58 A5 C6 D0 03 4C 81   39
0919 : EA AD 77 02 C9 11 D0 78   51
0921 : A5 D6 C9 18 D0 F0 A5 CF   71
0929 : F0 06 A9 00 85 C6 F0 E6   A7
0931 : A5 D1 85 5C A5 D2 85 5D   6D
0939 : A9 19 85 5E A0 01 20 E0   F0
0941 : C1 C9 3A F0 1D C6 5E D0   B7
0949 : 0C A9 00 85 5F 85 C6 20   58
0951 : 0A C2 4C 4C C1 38 A5 5C   86
0959 : E9 28 85 5C B0 DE C6 5D   1B
0961 : D0 DA 20 EA C1 B0 2E A9   B1
0969 : 00 85 C6 A5 5F 69 08 90   15
0971 : 06 20 5F C8 4C 5F C1 A8   90
0979 : 20 72 C8 98 20 D5 C7 20   28
0981 : 5C C8 A9 08 20 43 C8 20   2C
0989 : EA C8 A2 18 A0 01 20 0C   2E
0991 : E5 A9 05 85 CD 4C 81 EA   58
0999 : C9 91 D0 F9 A5 D6 D0 F5   DE
09A1 : A5 CF F0 06 A9 00 85 C6   69
09A9 : F0 EB A5 D1 85 5C A5 D2   AA
09B1 : 85 5D A9 19 85 5E A0 01   42
09B9 : 20 E0 C1 C9 3A F0 1F C6   28
09C1 : 5E D0 0E A9 00 85 C6 20   C8
09C9 : 20 C2 20 0A C2 A9 F8 D0   93
09D1 : 20 18 A5 5C 69 28 85 5C   99
09D9 : 90 DC E6 5D D0 D8 20 EA   67
09E1 : C1 B0 B2 A9 00 85 C6 20   64
09E9 : 20 C2 38 A5 5F E9 08 90   B4
09F1 : 13 A8 20 75 C8 98 20 D5   8D
09F9 : C7 20 5C C8 A9 08 20 43   E2
0A01 : C8 20 EA C8 A2 00 A0 01   5C
0A09 : 4C 63 C1 B1 5C C8 C9 20   21
0A11 : B0 02 09 40 60 C0 28 D0   5B
0A19 : 02 38 60 20 E0 C1 C9 20   D7
0A21 : F0 F3 20 21 C8 0A 0A 0A   50
0A29 : 0A 85 5F 20 E0 C1 20 21   B1
0A31 : C8 05 5F 85 5F A9 FF 85   53
0A39 : CC A5 CF F0 0A A5 CE A4   3C
0A41 : D3 91 D1 A9 00 85 CF A5   3D
0A49 : 5F 18 60 A6 D2 20 27 C2   F2
0A51 : A6 F4 E8 E8 E8 86 AD 86   4F
0A59 : 61 A2 00 86 AC A9 28 85   A0
0A61 : 60 A0 C0 A2 03 88 B1 AC   2A
0A69 : 91 60 98 D0 F8 C6 AD C6   74
0A71 : 61 CA 10 F1 A9 20 A6 D2   55
0A79 : 86 61 84 60 A0 27 91 60   27
0A81 : 88 10 FB A2 00 A0 00 20   AA
0A89 : 0C E5 60 20 CA C0 A9 80   FE
0A91 : 8D 71 C9 A9 0E 8D 20 D0   EE
0A99 : A9 06 8D 21 D0 A0 00 B9   52
0AA1 : 83 CA F0 06 20 D2 FF C8   B0
0AA9 : D0 F5 20 87 C2 68 68 4C   16
0AB1 : 7B E3 A9 92 8D 08 03 A9   53
0AB9 : C2 8D 09 03 60 20 73 00   B9
0AC1 : C9 44 F0 06 20 79 00 4C   10
0AC9 : E7 A7 A2 00 8E 72 C9 A0   11
0AD1 : 01 BD 60 CA F0 08 D1 7A   AE
0AD9 : D0 15 E8 C8 D0 F3 20 FB   AC
0AE1 : A8 AD 72 C9 0A AA BD 76   0F
0AE9 : CA 48 BD 75 CA 48 60 EE   44
0AF1 : 72 C9 AD 72 C9 C9 07 F0   EA
0AF9 : 0B BD 60 CA F0 03 E8 D0   C1
0B01 : F8 E8 D0 CB A0 00 F0 BC   62
0B09 : A9 01 2C A9 00 85 0A 20   08
0B11 : F6 C2 20 6F E1 4C AE A7   E9
0B19 : 20 F6 C2 20 59 E1 4C AE   9C
```

```
0B21 : A7 A9 00 20 BD FF A2 08   17
0B29 : A0 00 20 BA FF 20 06 E2   08
0B31 : 20 57 E2 60 20 00 E2 8A   64
0B39 : A8 A2 08 4C BA FF 20 75   D5
0B41 : C4 4C AE A7 4C F9 E0 A9   37
0B49 : 0F 20 C3 FF 20 2F C3 20   24
0B51 : C0 FF B0 F0 20 C3 FF 4C   14
0B59 : AE A7 A9 00 20 BD FF A9   88
0B61 : 0F A8 A2 08 20 BA FF 20   86
0B69 : 06 E2 20 57 E2 60 20 E9   59
0B71 : C4 4C AE A7 20 CF FF C9   10
0B79 : 0D F0 13 20 ED C7 90 3C   9F
0B81 : 8D 50 CA 20 39 C8 20 ED   23
0B89 : C7 90 31 8D 53 CA 20 6C   7B
0B91 : C3 20 59 C8 4C 75 C4 20   97
0B99 : 7C C8 AD 6C C9 C9 01 F0   43
0BA1 : 1C A9 31 20 B0 C8 A2 0D   D8
0BA9 : 20 C6 FF A2 00 20 CF FF   C1
0BB1 : 9D 00 CF E8 D0 F7 20 CC   46
0BB9 : FF 20 A3 C8 60 20 25 C9   F9
0BC1 : A2 0D 20 C9 FF A2 00 BD   BC
0BC9 : 00 CF 20 D2 FF E8 D0 F7   8E
0BD1 : 20 CC FF A9 32 20 B0 C8   05
0BD9 : 4C 8E C3 AE 00 CF F0 0C   8D
0BE1 : AD 01 CF 8E 50 CA 8D 53   0D
0BE9 : CA 4C 63 C3 A2 C5 A0 CA   9B
0BF1 : 4C 3A C9 AE 50 CA AD 53   5B
0BF9 : CA 18 69 01 DD AF C9 90   ED
0C01 : E2 A9 00 E8 E0 24 D0 DB   FF
0C09 : A2 01 D0 D7 AE 50 CA AD   4F
0C11 : 53 CA 38 E9 01 B0 CC CA   73
0C19 : D0 02 A2 23 BC AF C9 88   79
0C21 : 98 D0 C0 20 CF FF C9 0D   94
0C29 : F0 5A 20 ED C7 90 69 F0   95
0C31 : 67 C9 24 48 20 CA C0 A9   3E
0C39 : 80 8D 71 C9 68 8D 50 CA   DF
0C41 : A9 00 8D 53 CA A2 D5 A0   12
0C49 : CA 20 3A C9 AD 50 CA 20   B4
0C51 : D5 C7 20 5F C8 20 5F C8   9A
0C59 : AD 53 CA 20 D5 C7 20 59   35
0C61 : C8 20 4B C9 A9 08 A0 00   A3
0C69 : 20 43 C8 20 EA C8 20 5F   95
0C71 : C8 AE 50 CA AC 53 CA C8   20
0C79 : 98 DD AF C9 B0 06 8D 53   3D
0C81 : CA 4C 2D C4 A9 00 8D 00   26
0C89 : CF 8D 01 CF 8D 50 CA 8D   FB
0C91 : 53 CA 8D 71 C9 20 B3 C0   C9
0C99 : 60 20 CF FF C9 0D D0 4A   DA
0CA1 : 20 5F C8 A9 00 85 90 20   87
0CA9 : CA C0 A9 08 85 BA 20 B4   57
0CB1 : FF A9 6F 85 B9 20 96 FF   08
0CB9 : 20 A5 FF C9 30 D0 20 20   2F
0CC1 : A5 FF C9 30 D0 08 A9 91   F5
0CC9 : 20 D2 FF 4C B9 C4 48 A9   12
0CD1 : 30 20 D2 FF 68 4C B4 C4   0B
0CD9 : 20 A5 FF 24 90 70 05 20   31
0CE1 : D2 FF D0 F4 20 AB FF 4C   7D
0CE9 : B3 C0 C9 24 D0 03 4C E9   1D
0CF1 : C4 48 20 CA C0 A9 08 85   BF
0CF9 : BA 20 B1 FF A9 6F 85 B9   CF
```

```
0D01 : 20 93 FF 68 20 A8 FF 20   7F
0D09 : CF FF C9 0D D0 F6 20 AE   8E
0D11 : FF 4C 75 C4 20 5F C8 20   8D
0D19 : CA C0 A9 14 85 FD A9 00   1F
0D21 : 85 90 A9 3E 85 BB A9 CA   93
0D29 : 85 BC A9 01 85 B7 A9 08   64
0D31 : 85 BA A9 60 85 B9 20 D5   DC
0D39 : F3 20 7A C5 A4 90 D0 58   56
0D41 : A0 06 84 FB 20 A5 FF A6   01
0D49 : FC 85 FC A4 90 D0 49 A4   D9
0D51 : FB 88 D0 EE A4 FC 20 CD   F0
0D59 : BD A9 20 20 D2 FF 20 A5   F0
0D61 : FF A6 90 D0 33 AA F0 06   4A
0D69 : 20 D2 FF 4C 33 C5 20 5F   1C
0D71 : C8 C6 FD D0 1F 20 AB FF   D7
0D79 : A9 14 85 FD 20 B3 C0 A5   3B
0D81 : C5 C9 40 F0 FA 20 ED F6   AF
0D89 : F0 0E 20 7A C5 A4 90 D0   3D
0D91 : 07 20 CA C0 A0 04 D0 AA   36
0D99 : 20 42 F6 20 B3 C0 A9 00   84
0DA1 : 85 C6 4C 78 C4 A5 BA 20   50
0DA9 : B4 FF A5 B9 20 96 FF 60   75
0DB1 : 20 ED C7 90 13 A8 A9 08   F9
0DB9 : 85 97 20 39 C8 20 39 C8   3D
0DC1 : 20 19 C9 D0 F8 20 EA C8   C8
0DC9 : 60 A0 00 8C 6F C9 88 8C   8B
0DD1 : 70 C9 20 CF FF C9 0D F0   8C
0DD9 : 15 20 ED C7 90 10 8D 6F   11
0DE1 : C9 20 CF FF C9 0D F0 06   83
0DE9 : 20 ED C7 8D 70 C9 AC 6F   8A
0DF1 : C9 20 64 C8 20 72 C8 98   E6
0DF9 : 20 D5 C7 20 5C C8 A9 08   BC
0E01 : 20 43 C8 20 EA C8 4C C6   AD
0E09 : C5 20 CF FF C9 0D F0 22   DF
0E11 : C9 20 F0 F5 C9 54 F0 6D   C3
0E19 : C9 41 D0 03 4C 9F C6 20   34
0E21 : 02 C8 90 60 8D 5B CA 20   D7
0E29 : 39 C8 20 ED C7 90 55 8D   FD
0E31 : 5E CA A9 03 CD 6C C9 D0   C8
0E39 : 03 A9 41 2C A9 46 8D 57   98
0E41 : CA 20 CA C0 20 7C C8 AD   4A
0E49 : 5B CA 48 20 2C C8 8E 5B   19
0E51 : CA 8D 5C CA AD 5E CA 48   DC
0E59 : 20 2C C8 8E 5E CA 8D 5F   C4
0E61 : CA A2 0F 20 C9 FF A2 00   6B
0E69 : BD 55 CA 20 D2 FF E8 E0   1A
0E71 : 0B D0 F5 68 8D 5E CA 68   36
0E79 : 8D 5B CA 20 CC FF 20 A3   FF
0E81 : C8 20 66 C3 60 20 39 C8   E9
0E89 : 20 ED C7 C9 00 F0 F5 C9   BE
0E91 : 24 B0 F1 85 5F 20 C1 C6   C6
0E99 : A9 03 CD 6C C9 D0 03 18   24
0EA1 : 90 01 38 08 A5 5F 48 20   78
0EA9 : CE C6 68 68 A2 01 8E 6C   47
0EB1 : C9 20 63 C3 A2 01 86 C6   B5
0EB9 : A2 0D 8E 77 02 A9 49 20   47
0EC1 : C6 C4 A2 00 8E 71 C9 4C   C6
0EC9 : F6 CA 20 C1 C6 A9 23 85   B6
0ED1 : FB A9 03 CD 6C C9 D0 03   7A
0ED9 : 18 90 01 38 08 A5 FB 48   AF
```

```
0EE1 : 20 CE C6 68 C6 FB D0 F5  A2
0EE9 : 48 4C 7F C6 20 CA C0 A2  B1
0EF1 : 12 A9 00 8E 71 C9 4C B8  B2
0EF9 : C3 BA E8 E8 9A 68 AA BC  82
0F01 : AF C9 0A 0A AA 28 B0 04  0F
0F03 : A9 00 F0 19 98 9D 00 CF  28
0F11 : A9 FF 9D 01 CF 9D 02 CF  D3
0F19 : 98 38 E9 11 A8 B9 D3 C9  A5
0F21 : 9D 03 CF D0 0F 9D 00 CF  CB
0F29 : 9D 01 CF 9D 02 CF 9D 03  09
0F31 : CF F0 01 68 BA CA CA CA  88
0F39 : CA 9A 60 20 CF FF C9 0D  AB
0F41 : F0 22 20 3A C7 20 CA C0  BC
0F49 : 78 A9 35 85 01 B1 FB 99  55
0F51 : 00 CF C8 D0 F8 A9 37 85  4A
0F59 : 01 20 B3 C0 A9 00 8D 50  E1
0F61 : CA 8D 53 CA 60 20 39 C8  9D
0F69 : 20 02 C8 85 FC 20 39 C8  B4
0F71 : 20 02 C8 85 FB A0 00 60  FB
0F79 : 20 CF FF C9 0D F0 E5 20  EA
0F81 : 3A C7 B9 00 CF 91 FB C8  18
0F89 : D0 F8 4C 31 C7 A9 04 AA  3E
0F91 : A0 00 20 BA FF A9 00 20  1E
0F99 : BD FF 20 C0 FF A2 04 A2  E0
0FA1 : 04 20 C9 FF 4C CA C0 20  86
0FA9 : 5F C8 20 5F C8 A2 E0 A0  C7
0FB1 : CA 20 3A C9 AD 50 CA 20  1C
0FB9 : D5 C7 A2 EB A0 CA 20 3A  ED
0FC1 : C9 AD 53 CA 20 D5 C7 4C  F7
0FC9 : 5F C8 20 CF FF C9 0D F0  F3
0FD1 : 09 C9 24 F0 29 C9 42 F0  B2
0FD9 : 26 60 20 C0 C7 20 5F C8  DC
0FE1 : 20 CC FF A9 04 20 C3 FF  ED
0FE9 : 4C B3 C0 A0 FF 8C 70 C9  0D
0FF1 : C8 8C 6F C9 20 62 C7 20  89
0FF9 : 7C C7 20 C3 C5 60 60 60  7B
1001 : 48 4A 4A 4A 4A 20 17 C8  DE
1009 : AA 68 29 0F 20 17 C8 48  82
1011 : 8A 20 D2 FF 68 4C D2 FF  94
1019 : A9 00 8D 6E C9 20 39 C8  08
1021 : C9 20 D0 09 20 39 C8 C9  D2
1029 : 20 D0 0F 18 60 20 21 C8  95
1031 : 0A 0A 0A 0A 8D 6E C9 20  B8
1039 : 39 C8 20 21 C8 0D 6E C9  45
1041 : 38 60 18 69 F6 90 02 69  AB
1049 : 06 69 3A 60 C9 3A 08 29  7F
1051 : 0F 28 90 02 69 08 60 A2  76
1059 : 30 38 E9 0A 90 03 E8 B0  87
1061 : F9 69 3A 60 20 CF FF C9  BE
1069 : 0D D0 02 68 68 60 85 97  3B
1071 : 20 5C C8 B9 00 CF 20 D5  D3
1079 : C7 C8 D0 03 EE 6D C9 C6  48
1081 : 97 D0 ED 60 20 5C C8 A9  63
1089 : 20 2C A9 0D 4C D2 FF AD  82
1091 : 6D C9 D0 06 CC 70 C9 B0  B1
1099 : 01 60 68 68 60 20 5F C8  08
10A1 : A9 3A A2 2E 4C E4 C7 20  21
10A9 : CA C0 A9 0F A8 A2 08 20  20
10B1 : BA FF A9 00 20 BD FF 20  05
10B9 : C0 FF A9 0D A8 A2 08 20  85
10C1 : BA FF A9 01 A2 3F A0 CA  42
10C9 : 20 BD FF 4C C0 FF A9 0D  1E
10D1 : 20 C3 FF A9 0F 20 C3 FF  09
10D9 : 4C B3 C0 8D 49 CA AD 50  23
10E1 : CA 48 20 2C C8 8E 50 CA  35
10E9 : 8D 51 CA AD 53 CA 48 20  74
10F1 : 2C C8 8E 53 CA 8D 54 CA  8F
10F9 : A2 0F 20 C9 FF A2 00 BD  F5
1101 : 48 CA 20 D2 FF E8 E0 0D  F6
1109 : D0 F5 68 8D 53 CA 68 8D  E8
1111 : 50 CA 4C CC FF 98 38 E9  EC
1119 : 08 A8 20 5C C8 A9 12 20  6B
1121 : D2 FF A2 08 B9 00 CF 29  CA
1131 : 03 B9 00 CF 20 D2 FF A9  F7
1139 : 00 85 D4 C8 CA D0 E5 A9  68
1141 : 92 4C D2 FF 20 ED C7 90  60
1149 : 03 99 00 CF C8 C6 97 60  F5
1151 : A2 0F 20 C9 FF A2 00 BD  4D
1159 : 40 CA 20 D2 FF E8 E0 08  3C
1161 : D0 F5 4C CC FF 86 FB 84  06
1169 : FC A0 00 B1 FB F0 06 20  8B
1171 : D2 FF C8 D0 F6 60 20 7C  7B
1179 : C8 A9 31 20 B0 C8 A2 0D  5C
1181 : 20 C6 FF A2 00 20 CF FF  99
1189 : 9D 00 CF E8 E0 08 D0 F5  B5
1191 : 20 CC FF 4C A3 C8 FF 0D  3B
1199 : 00 00 00 FF 80 04 3A 57  59
11A1 : 52 41 46 4E 40 24 4D 47  D8
11A9 : 50 53 43 58 42 55 48 2B  C5
11B1 : 2D 54 C5 C3 C3 C5 C5 C3  FB
11B9 : C4 C4 C5 C7 C7 C7 C7 C2  A9
11C1 : CA BF BF C3 C3 C3 84 48  D0
11C9 : 48 DD DD AF 6D E8 9D 0F  20
11D1 : 4C 7B 9E 5F F5 FF FF C7  5D
11D9 : E0 F7 00 15 15 15 15 15  D0
11E1 : 15 15 15 15 15 15 15 15  E1
11E9 : 15 15 15 15 13 13 13 13  AD
11F1 : 13 13 13 12 12 12 12 12  B3
11F9 : 12 11 11 11 11 11 01 03  9E
1201 : 07 1F 1F 93 1F 20 12 20  4E
1209 : 20 20 20 20 20 20 20 31  2B
1211 : 35 34 31 20 44 49 53 4B  23
1219 : 2D 4D 4F 4E 49 54 4F 52  A4
1221 : 20 56 20 32 2E 35 20 20  08
1229 : 20 20 20 20 20 0D 0D 20  44
1231 : 20 20 20 20 20 20 20 20  31
1239 : 20 20 20 20 20 20 20 20  39
1241 : 20 20 20 20 20 20 20 20  41
1249 : 20 20 20 20 20 0D 20 20  B1
1251 : 20 20 20 20 20 20 20 20  51
1259 : 20 20 20 20 20 20 20 20  59
1261 : 20 20 20 20 20 20 20 0D  3B
1269 : 00 24 23 42 2D 50 20 31  C5
1271 : 33 20 30 55 31 3A 31 33  7B
1279 : 20 30 20 12 38 20 00 30  E1
1281 : 42 2D 41 20 30 20 00 00  B2
1289 : 20 00 00 93 00 94 00 95  EC
1291 : 00 49 53 4B 00 49 52 00  08
1299 : 45 52 52 B0 00 4D 91 00  63
12A1 : DF C2 EC C2 DC C2 1B C3  4D
```

```
12A9 : 42 C3 12 C3 FF BF 93 9A  4B
12B1 : 12 44 4C 4F 41 44 20 20  D9
12B9 : 0D 12 44 53 41 56 45 20  67
12C1 : 20 0D 12 44 56 45 52 49  E0
12C9 : 46 59 0D 12 44 49 52 20  5A
12D1 : 20 20 20 0D 12 44 49 53  BA
12D9 : 4B 20 20 20 0D 12 44 45  3D
12E1 : 52 52 4F 52 20 0D 12 44  B6
12E9 : 4D 4F 4E 20 20 20 0D 00  AD
12F1 : 20 20 45 4E 44 20 4F 46  4C
12F9 : 20 46 49 4C 45 20 21 00  F2
1301 : 93 20 54 52 41 43 4B 20  9F
1309 : 3A 20 00 20 20 54 52 41  C8
1311 : 43 4B 20 3A 20 00 20 53  72
1319 : 45 4B 54 4F 52 20 3A 20  52
1321 : 00 20 04 CB A2 00 8E 71  F3
1329 : C9 20 B3 C0 4C 1E CC 20  30
1331 : C1 C6 A2 29 A0 CC 20 3A  88
1339 : C9 A2 90 BD 00 CF 20 D2  D4
1341 : FF E8 E0 A0 D0 F5 20 59  F0
1349 : C8 A2 60 A0 CC 20 3A C9  D9
1351 : AD A2 CF 20 D2 FF AD A3  72
1359 : CF 20 D2 FF 20 5F C8 20  4D
1361 : 5F C8 20 5F C8 A0 14 98  2B
1369 : 20 D5 C7 A9 2D 20 D2 FF  BA
1371 : 98 F0 03 20 5F C8 88 98  D6
1379 : 10 EE A2 02 A0 03 20 0C  A4
1381 : E5 A0 03 A9 30 A2 0F D0  A2
1389 : 0A C0 01 F0 04 A2 10 D0  89
1391 : 02 A2 04 20 D2 FF CA D0  E3
1399 : FA 18 69 01 88 D0 EA A2  1A
13A1 : 03 A0 03 20 0C E5 A0 23  72
```

```
13A9 : A9 01 85 FB 20 17 C8 20  D2
13B1 : D2 FF E6 FB A5 FB C9 10  3D
13B9 : D0 04 A9 00 85 FB 88 D0  F2
13C1 : EB A9 01 85 02 A2 04 A5  02
13C9 : 02 69 02 A8 20 0C E5 A2  55
13D1 : 14 86 8B A9 CF 85 8D A9  F3
13D9 : 01 A6 02 69 04 CA D0 FB  AD
13E1 : 85 8C A5 8B A6 02 E0 12  A9
13E9 : 90 20 E0 19 90 14 E0 1F  50
13F1 : 90 08 C9 11 90 14 A2 04  56
13F9 : D0 2A C9 12 90 0C A2 04  8F
1401 : D0 22 C9 13 90 04 A2 04  73
1409 : D0 1A A5 8B 4A 4A 4A A8  33
1411 : B1 8C A8 A5 8B 29 07 AA  5B
1419 : 98 3D 66 CC D0 04 A2 00  3B
1421 : F0 02 A2 02 A0 02 BD 6E  E9
1429 : CC 20 D2 FF E8 88 D0 F6  BE
1431 : A5 8B F0 05 A9 11 20 D2  C2
1439 : FF C6 8B 10 A5 E6 02 A5  65
1441 : 02 C9 24 F0 03 4C 9A CB  E4
1449 : 60 A9 00 85 C6 A5 C6 F0  C5
1451 : FC C6 C6 60 93 20 20 2A  7D
1459 : 3D 44 4F 4C 55 20 20 20  2D
1461 : 20 2E 3D 42 4F 53 20 20  80
1469 : 20 A6 3D 59 4F 4B 20 20  67
1471 : 20 20 20 20 20 20 20 20  71
1479 : 20 20 20 20 20 20 20 44  C2
1481 : 49 53 4B 20 41 44 49 20  E6
1489 : 3A 20 00 49 44 20 3A 20  6B
1491 : 00 01 02 04 08 10 20 40  15
1499 : 80 2A 9D 2E 9D A6 9D 60  A2
```

Pascal (9)

PROGRAM 1

```
program sıcaklık;

var
   ısı :array[1..24] of real;
   total,averaj,yuksek,alcak:real;
   ı :integer;

begın
   (* verıleri okuma *)
   wrıte('24 sıcaklık gırın.');
   for ı := 1 to 24 do
       readln(ısı[ı]);
   wrıteln;

   (* averaj ısıyı hesapla *)
   total := 0.0;
   for ı := 1 to 24 do
       total := total + ısı[ı];
   averaj := total / 24.0;
   wrıteln('averaj ısı: ',averaj:6:1);

   (* yuksek ve alcak ısıların hesaplanması  *)
   alcak := ısı[ı];
   yuksek := ısı[ı];
   for ı := 2 to 24  do
       ıf ısı[ı] < alcak then
          alcak := ısı[ı]
       else ıf ısı[ı] > yuksek then
          yuksek := ısı[ı];
   wrıteln('Dusuk ısı : ',alcak:6:1);
   wrıteln('Yuksek ısı : ',yuksek:6:1);
end.
```

PROGRAM 2

```
program fark;
const
   maksımum = 8;

type
   arr = array [1..maksımum] of real;

var
   x : arr;
   averaj,
   toplam :real;
   ı : integer;
```

```
begin
  (* veri girisi *)
  write(maksimum,' sayi girin');
  for i := 1 to maksimum do
      readln(x[i]);
  readln;

  (* averaj degeri hesapla  *)
  toplam := 0.0;
  for i := 1 to maksimum do
      toplam := toplam + x[i];
  averaj  := toplam / maksimum;
  writeln(' Averaj deger = ',averaj:8:1);

  (* averajla sayi arasindaki farki bul *)
  writeln('Averajla sayi arasindaki fark');
  writeln('i':4,'x[i]':8,'Fark ':14);
  for i := 1 to maksimum do
      writeln(i:4,x[i]:8:1,x[i]-averaj:14:1);
end.
```

PROGRAM 3

```
program hesap;
const
  sinifno = 10;

type
  indeks = 1..sinifno;
  hesarr = array[indeks] of real;

var
  butce : hesarr;

procedure hazirla(var butce : hesarr);
var
  sonraki : indeks;

begin
  for sonraki  := 1 to sinifno  do
      butce[sonraki] := 0.0
end;
procedure hesapla(var butce : hesarr);
const
  son = 0;

type
  si = 0..sinifno;

var
  sinif : si;
  harcama : real;

procedure oku(var sinif : si; var harcama  : real);
var
  gecici : integer;

begin
  repeat
   write(sinifno:2,'ile 1 arasinda hesap numarasi yazin. Veya 0 durmak icin');
```

Commodore

```
    readln(gecici);
  until gecici in [0..sinifno];
  sinif := gecici;
  if sinif<>son then
    begin
      write('Hesap miktarini girin $ ');
      readln(harcama)
    end
end;

begin (* oku *)
  oku(sinif,harcama);
  while sinif <> son do
    begin
      butce[sinif] := butce[sinif] + harcama;
      oku(sinif,harcama)
    end
end;  (* hesapla *)

procedure rapor(butce : hesarr);
var
  sonraki : indeks;

begin
  writeln('Sinifi':10,'Harcama':15);
  for sonraki := 1 to sinifno do
      writeln(sonraki:10,butce[sonraki]:15:2)
end;

begin
  hazirla(butce);
  hesapla(butce);
  rapor(butce);
end.
```

PROGRAM 4

```
program say1;
type
  harf = 'a'..'z';
var
  say : array [harf] of integer;
  bul : array [harf] of boolean;
  sonraki:char;

begin
  for sonraki := 'a' to 'z' do
      begin
        say[sonraki] := 0;
        bul[sonraki] := false;
      end;

(* satirdaki harfleri say *)
writeln('Bir satir yazi yazin');
while not eoln do
   begin
     readln(sonraki);
     if sonraki in ['a'..'z'] then
        begin
          say[sonraki] := say[sonraki]+1;
```

```
            bul[sonraki] := true;
          end;
    end;
 writeln;
 writeln('harf = ','Sayi = ':16);
 for sonraki := 'a' to 'z' do
    if bul[sonraki] then
       writeln(sonraki:6,say[sonraki]:16)
end.
```

PROGRAM 5

```
 program sifre(input,output);
 type
   harf = 'a'..'z';
   kodarr = array[harf]  of char;

 var
   kod : kodarr;

 procedure kodoku(var kod : kodarr);
 var
   sharf : harf;

 begin
   writeln('Her harfin yanina bir kod girin.');
  for sharf := 'a' to 'z' do
     begin
       write(sharf:4);
       readln(kod[sharf]);
     end;
   writeln;
 end; (* kodoku *)

procedure yaz(kod : kodarr);
 const
    son = '-';

 var
    sonraki : char;

 begin
   writeln('Mesajinizi yazmayi "-" ile bitirin.');
   repeat
     read(sonraki);
     if sonraki in ['a'..'z'] then
        writeln(kod[sonraki]:5)
     else
        writeln(sonraki:5)
  until sonraki = son
end; (* yaz *)

begin (* sifre *)
   kodoku(kod);
   yaz(kod)
 end.
```

C 128 Püf Noktaları'nın Programları

PROGRAM 1

```
100   FOR X = 6656 TO 6703
110   READ A: CS = CS + A:  POKE X,A
120   NEXT
130   DATA 169,0,141,0,255,169,0,32,183,255
140   DATA 169,4,170,160,7,32,186,255,32,192
150   DATA 255,162,4,32,201,255,165,45,133,37
160   DATA 165,46,133,98,32,228,80,169,0,141
170   DATA 0,255,169,4,32,231,255,96
```

PROGRAM 2

```
10 REM UYARI SESI
20 FOR I=5165 TO 5234
30 READ A
40 S=S+A
50 POKE I,A
60 NEXT I
70 IF S<> 8167 THEN BEGIN
80 PRINT "?HATA VAR"
90 END
100 BEND
110 SYS 5165
120 DATA 169,255,141,6,212,141,24,212
130 DATA 169,9,141,5,212,120,169,71
140 DATA 141,20,3,169,20,141,21,3,88,96
150 DATA 72,165,231,229,236,201,2,208
160 DATA 17,205,128,20,240,15,141,128
170 DATA 20,169,160,141,1,212,169,33
180 DATA 206,5,141,128,20,169,0,141
190 DATA 1,212,141,0,212,141,4,212
200 DATA 104,76,101,250
```

PROGRAM 3

```
10 TRAP 5000
20 SCNCLR
100 PRINT"■LUTFEN BEKLEYIN..."
110 DIM TK$(255),FE$(29),CE$(8),SZ$(153)
120 FOR Z=0 TO 125:READ WE$,A$:TK$(DEC(WE$))=A$:NEXT Z
130 FOR Z=0 TO 27:READ WE$,A$:FE$(DEC(WE$))=A$:NEXT Z
140 FOR Z=0 TO 7: READ WE$,A$:CE$(DEC(WE$))=A$:NEXT Z
150 FOR Z=0 TO 25:READ WE$,A$:SZ$(DEC(WE$))=A$:NEXT Z
160 AN = 7168
170 PRINT "■'E'KRAN VEYA 'Y'AZICI ?"
```

```
180 DR = 0:GET KEY A$:IF A$ = "Y" THEN DR=1:GOTO 200
190 IF A$<>"E" THEN 170
200 PRINT "■'B'UYUK VEYA 'K'UCUK HARF ?"
210 GET KEY A$:IF A$="K" THEN PRINT CHR$(14):GOTO 230
220 IF A$<>"B" THEN PRINT "□";:GOTO200
230 INPUT "
PROGRAM ISMI";NA$
240 IF DR THEN CLOSE 4:OPEN 4,4,0
250 PRINT : PRINT NA$:PRINT
260 IF DR THEN PRINT#4:PRINT#4,NA$:PRINT#4
270 FL=0:AN = AN +3
280 LO=PEEK(AN):AN=AN+1:HI=PEEK(AN):ZE=LO+256*HI
290 PRINT ZE;" ";
300 IF DR THEN PRINT#4,ZE;" ";
310 AN=AN+1
320 WE=PEEK(AN):IF WE=0 THEN 410
330 IF WE=32 THEN 460
340 IF WE=34 THEN 300
350 IF WE=58 THEN 360
360 IF WE = DEC("FE") THEN 520
370 IF WE = DEC("CE") THEN 580
380 IF FF THEN 740
390 GOTO 640
400 :
410 REM YENI SATIR
420 IF DR THEN PRINT#4
430 PRINT:IF PEEK(AN+1)=0 AND PEEK(AN+2)=0 THEN CLOSE 4:END
440 GOTO 270
450 :
460 REM BOS SATIR
470 IF FF THEN 830
480 IF FL THEN PRINT " ";
490 IF DR AND FL THEN PRINT#4," ";
500 GOTO 310
510 :
520 REM $FE
530 AN=AN+1:WE=PEEK(AN)
540 PRINT FE$(WE);" ";
550 IF DR THEN PRINT#4,FF$(WE);" ";
560 GOTO 310
570 :
580 REM $CE
590 AN=AN+1:WE = PEEK(AN)
600 PRINT CE$(WE);" ";
610 IF DR THEN PRINT#4,CE$(WE);" ";
620 GOTO 310
630 :
640 REM KAREKTER DIZI BASLANGICI
650 IF WE=143 THEN FL = 1
660 IF TK$(WE)<> "" THEN 700
670 PRINT CHR$(WE);
680 IF DR THEN PRINT#4,CHR$(WE);
690 GOTO 310
700 PRINT " ";TK$(WE);" ";
710 IF DR THEN PRINT#4," ";TK$(WE);" ";
720 GOTO 310
730 :
740 REM KAREKTER DIZILERININ IC KISIMLARI
750 IF WE<> 255 THEN 790
760 PRINT"▚";
770 IF DR THEN PRINT#4,"▚";
780 GOTO 310
790 IF SZ$(WE)<>"" THEN 830
```

```
800 PRINT CHR$(WE);
810 IF DR THEN PRINT#4,CHR$(WE);
820 GOTO 310
830 ZA=1

840 IF PEEK(AN+1)<> WE THEN 860
850 AN=AN+1:ZA=ZA+1:GOTO840
860 PRINT CHR$(18);"[";SZ$(WE);STR$(ZA);"]";CHR$(146);
870 IF DR THEN PRINT#4," [";SZ$(WE);STR$(ZA);"] ";
880 GOTO 310

890 :
900 REM TIRNAK ISARETI
910 FF=XOR(FF,255)
920 PRINT CHR$(34);: POKE 244,0
930 IF DR THEN PRINT#4,CHR$(34);
940 GOTO 310
950 :

960 REM KOLON
970 PRINT CHR$(58);" ";
980 IF DR THEN PRINT#4,CHR$(58);" ";
990 GOTO 310

1000 :
1010 DATA "80","END","81","FOR","82","NEXT","83","DATA","84","INPUT#"
1020 DATA "85",INPUT","86","DIM","87","READ","88","LET","89","GOTO"
1030 DATA "8A","RUN","8B","IF","8C","RESTORE","3D","GOSUB","3E","RETURN"
1040 DATA "8F","REM","90","STOP","91","ON","92","WAIT","93","LOAD","94","SAVE"
1050 DATA "95","VERIFY","96","DEF","97","POKE","98","PRINT#","99","PRINT"
1060 DATA "9A","CONT","9B","LIST","9C","CLR","9D","CMD","9E","SYS","9F","OPEN"
1070 DATA "A0","CLOSE","A1","GET","A2","NEW","A3","TAB(","A4","TO","A5","FN"
1080 DATA "A6","SPC(","A7","THEN","A8","NOT","A9","STEP","AA","+","AB","-"
1090 DATA "AC","*","AD","/","AE","↑","AF","AND","B0","OR","B1",">","B2","="
1100 DATA "B3","<","B4","SGN","B5","INT","B6","ABS","37","USR","B8","FRE"
1110 DATA "B9","POS","BA","SQR","BB","RND","BC","LOG","BD","EXP","BE","COS"
1120 DATA "BF","SIN","C0","TAN","C1","ATN","C2","PEEK","C3","LEN","C4","STR$"
1130 DATA "C5","VAL","C6","ASC","C7","CHR$","C8","LEFT$","C9","RIGHT$"
1140 DATA "CA","MID$","CB","GO","CC","RGR","CD","RCLR","CF","JOY","D0","RDOT"
1150 DATA "D1","DEC","D2","HEX$","D3","ERR$","D4","INSTR","D5","ELSE"
1160 DATA "D6","RESUME","D7","TRAP","D8","TRON","D9","TROFF","DA","SOUND"
1170 DATA "DB","VOL","DC","AUTO","DD","PUDEF","DE","GRAPHIC","DF","PAINT"
1180 DATA "E0","CHAR","E1","BOX","E2","CIRCLE","E3","GSHAPE","E4","SSHAPE"
1190 DATA "E5","DRAW","E6","LOCATE","E7","COLOR","E8","SCNCLR","E9","SCALE"
1200 DATA "EA","HELP","EB","DO","EC","LOOP","ED","EXIT","EE","DIRECTORY"
1210 DATA "EF","DSAVE","F0","DLOAD","F1","HEADER","F2","SCRATCH"
1220 DATA "F3","COLLECT","F4","COPY","F5","RENAME","F6","BACKUP","F7","DELETE"
1230 DATA "F8","RENUMBER","F9","KEY","FA","MONITOR","FB","USING","FC","WHILE"
1240 DATA "FD","UNTIL","FF","π"

1250 DATA "02","BANK","03","FILTER","04","PLAY","05","TEMPO","06","MOVSPR"
1260 DATA "07","SPRITE","08","SPRCOLOR","09","RREG","0A","ENVELOPE"
1270 DATA "0B","SLEEP","0C","CATALOG","0D","DOPEN","0E","APPEND","0F","DCLOSE"
1280 DATA "10","BSAVE","11","BLOAD","12","RECORD","13","CONCAT","14","DVERIFY"
1290 DATA "15","DCLEAR","16","SPRSAV","17","COLLISION","18","BEGIN"
1300 DATA "19","BEND","1A","WINDOW","1B","BOOT","1C","WIDTH","1D","SPRDEF"
1310 DATA "00","RWINDOW","02","POT","03","BUMP","04","PEN","05","RSPPOS"
1320 DATA "06","RSPRITE","07","RSPCOLOR","08","XOR"
1330 DATA "03","RUN STOP","05","WHT","0A","LINE FEED","11","CRSR DOWN"
1340 DATA "12","RVS ON","13","HOME","1C","RED","1D","CRSR RIGHT","1E","GRN"
1350 DATA "1F","BLU","20","SPACE","81","ORNG","90","BLK","91","CRSR UP"
1360 DATA "92","RVS OFF","93","CLR HOME","95","BRN","96","L RED"
1370 DATA "97","D GRY","98","M GRY","99","L GRN","9A","L BLU"
1380 DATA "9B","L GRY","9C","PUR","9D","CRSR LEFT","9E","YEL","9F","CYN"
5000 IF ER=5 THEN PRINT CHR$(19);"LUTFEN DISKET SURUCU VE YAZICIYI ACIN.":RESUME
5010 END
```

PROGRAM 4

```
20 REM --- 0 ---
30 DATA 31,248,,63,252,
40 DATA 127,254,,255,255,,240,15,
50 DATA 224,7,,224,7,,224,7,,224,7,
60 DATA 224,7,,224,7,,224,7,,224,7,
70 DATA 224,7,,224,7,,224,7,,224,7,
80 DATA 240,15,,255,255,,127,254,
90 DATA 63,252,,
100 REM --- 1 ---
110 DATA 3,252,,7,254,,7,254,
120 DATA 7,254,,7,254,,3,254,
130 DATA ,126,,,126,,,126,,,126,,,126,
140 DATA ,126,,,126,,,126,,,126,,,126,
150 DATA ,126,,,126,,,126,,,126,,,60,,
160 REM --- 2 ---
170 DATA 63,254,,127,255,,127,255,
180 DATA 124,31,,120,15,,48,15,,,15,
190 DATA ,31,,,63,,,126,,,252,,1,248,
200 DATA 3,240,,7,224,,15,160,,31,134,
210 DATA 63,7,,126,7,,252,7,,255,255,
220 DATA 127,254,,
230 REM --- 3 ---
240 DATA 63,254,,127,255,,127,255,
250 DATA 124,31,,120,15,,48,15,,,31,
260 DATA ,63,,,127,,,255,,1,255,
270 DATA 1,255,,,255,
280 DATA ,127,,,63,,96,31,,240,15,
290 DATA 240,15,,255,255,
300 DATA 255,255,,127,254,,
310 REM --- 4 ---
320 DATA ,124,,,252,,1,252,,1,252,
330 DATA 3,252,,3,252,,7,252,,7,252,
340 DATA 15,188,,15,188,,31,60,,31,60,
350 DATA 62,60,,124,60,,248,60,
360 DATA 255,255,,255,255,,255,255,
370 DATA 255,255,,,60,,,60,,
380 REM --- 5 ---
390 DATA 127,252,,255,255,,255,255,
400 DATA 240,15,,224,15,,224,6,,240,,
410 DATA 255,,,255,224,,127,248,
420 DATA 1,254,,,127,,,31,,,31,,,31,
430 DATA ,31,,96,127,,241,254,,255,248,
440 DATA 255,224,,127,128,,
450 REM --- 6 ---
460 DATA 127,252,,255,255,,255,255,
470 DATA 240,15,,224,15,,224,6,,240,,
480 DATA 255,,,255,224,,255,248,248,
490 DATA 254,,240,127,,224,31,,224,31,,224,31,
500 DATA 224,31,,240,127,,248,254,,255,248,
510 DATA 255,224,,127,128,,
520 REM --- 7 ---
530 DATA 63,254,,127,255,,127,255,
540 DATA 124,31,,120,15,,48,15,,,15,
550 DATA ,31,,,63,,,126,,,252,,1,248,
560 DATA 3,240,,7,224,,15,160,,31,128,
570 DATA 63,,,126,,,252,,,252,,,252,,,
580 REM --- 8 ---
590 DATA 127,254,,255,255,,255,255,
600 DATA 240,15,,224,7,,240,15,
610 DATA 255,255,,255,255,,127,254,
```

```
620 DATA 127,254,,254,127,,240,15,
630 DATA 224,7,,224,7,,224,7,,224,7,
640 DATA 240,15,,127,254,,31,248,
650 DATA 15,224,,1,128,,
660 REM --- 9 ---
670 DATA 1,254,,7,255,,31,255,,127,143,
680 DATA 254,15,,240,15,,240,15,
690 DATA 240,15,,240,15,,254,15,
700 DATA 127,143,,31,254,,7,255,,1,255,
710 DATA ,15,,96,7,,240,7,,240,15,
720 DATA 255,255,,255,255,,127,254,,
730 POKE 56334,PEEK(56334) OR 128
740 FOR I=3456 TO 4095
750 READ A
760 POKE I,A
770 NEXT I
780 REM ZAMAN KURULUYOR
780 INPUT "SAAT    ";S
800 IF S>12 THEN S=S-12:GOTO 800
810 IF S<0 THEN 790
820 POKE 56331,S + INT(S/10)*6
830 INPUT "DAKIKA ";M
840 IF M<0 OR M>59 THEN 830
850 POKE 56330,M+INT(M/10)*6
860 INPUT "SANIYE ";S
870 IF S<0 OR S>59 THEN 860
880 POKE 56329,S+INT(S/10)*6
890 S=0
900 FOR I=5120 TO 5180
910 READ A
920 S=S+A
930 POKE I,A
940 NEXT I
950 IF S<> 6300 THEN PRINT "?HATA VAR.":END
955 SCNCLR
960 PRINT "■BASLAMAK ICIN RETURN 'E BASIN"
965 GET A$:IF A$<> CHR$(13) THEN 965
970 POKE 56328,0
980 SYS 5120
990 DATA 120,169,13,141,20,3,169
1000 DATA 20,141,21,3,88,96,160
1010 DATA 3,173,11,220,41,31,75
1020 DATA 26,20,185,8,220,41,240
1030 DATA 74,74,74,74
1040 DATA 72,185,8,220,41,15
1050 DATA 72,136,208,237,173,8
1060 DATA 220,72,160,7,104
1070 DATA 24,105,54,153,247,7,136
1080 DATA 208,246,76,101,250
1090 FOR I=1 TO 7
1100 SPRITE I,1,1,1,0,1,0
1110 MOVSPR I,30+I*20+INT((I-1)/2)*10,100
1120 NEXT I
```